DASHUR CREDIT LIGHT TRANSFER WALLET INTEGRATION V1.6

2018-03-05

Revisions Log:

Date	Version	By	Action
23/11/2016	1.0	Nike	Initialized document contents.
06/12/2016	1.1	Jordan	Added parameter include_transfer to **Transaction Feed API**
23/8/2017	1.2	Jordan	Change account username length in **Create Member API**
12/10/2017	1.3	Jordan	Added new info for HTML5 Demo in **Launch Item API**
02/02/2018	1.4	Jordan	Added new column for Titanium LD game launch in **Launch Item API**
06/02/2018	1.5	Jordan	Added new option for Titanium LD game launch in **Launch Item API**
05/03/2018	1.6	Jordan	Added new columns to perform the redirection for the lobby/banking of HTML5 slot game and logout/failed for HTML5 Diamond Live Dealer game in **Launch Item API**

INTRODUCTION

This document describes the API and processes to integrate via transfer wallet where the wallet is a credit light wallet.

Each section describes one API needed and contains

1. A short description of the API
2. The url and method used
3. All request parameters with type and description
4. All response parameters with type and description
5. Examples including a curl example.

Before looking at each individual call make sure to read the definitions, especially with regards to the Timestamp type and Money type that Dashur uses.

DEFINTIONS

ITEM	DEFINITION
Auth Username	The username to use when calling the auth service. *This will be provided*
Auth Secret	The password for the auth service. *This will be provided*
API Username	The username of the user calling the API's. *This will be provided*
API Password	The password of the user calling the API's. *This will be provided*
Base Account ID	This is the base account id, from the id all child accounts can be found by traversing the API. *This will be provided*
Base URL	The base url of the APIs. *This will be provided*
Currency Codes	Dashur uses ISO three character currency codes Example USD, CNY, GBP, EUR See all supported currency codes in Appendix B
Error Response	There are two types of error responses. One for a general error and one for errors when validating input data. **Example General Error** `{ "meta": { "currency": "USD", "time_zone": "UTC", "transaction_id": "XXXX-XXX", "processing_time": -1 } "error": { "type": "HTTP_EXCEPTION", "code": "403", "message": "Forbidden", } }`

ITEM	DEFINITION

	Example Validation Error
	
	`{`
	`  “meta”: {`
	`  "currency": "USD",`
	`  "time_zone": "UTC",`
	`  "transaction_id": "XXXX-XXX",`
	`  "processing_time": -1`
	`  }`
	`  “error”: {`
	`  “type”: “HTTP_EXCEPTION”,`
	`  “code”: “403”,`
	`  “message”: “Forbidden”,`
	`  “fields”: [`
	`  {`
	`  “field”: “field1”,`
	`  “message”: “The reason for failure”,`
	`  }`
	`  ]`
	`  }`
	`}`

Headers	These are headers that are used to configure currency, time zone and locale for the response from Dashur. They need to be provided in each API Call. Additionally an Authorization header and an Accept header needs to be provided **Authorizaion –** This header is used to send the token to the API and is then used for authentication purposes **Accept** – The Accept header is used to indicate what format the caller wants the response in. Currently only “application/json” is supported **X-DAS-TZ** – This is the time zone used for dates requested and responded. All timezones are accepted in the following format: UTC, UTC+8, UTC-2 **X-DAS-CURRENCY** – This is the currency that will be used as default currency when querying different APIs. Most currencies are supported by using the ISO code for the currency (USD, CNY etc)

ITEM	DEFINITION

	X-DAS-LANG – The Language used in responses. Currently en and zh-CN are supported. **The language codes are case sensitive**
	
	X-DAS-TX-ID – A unique ID of the request. This is used to be able to trace transactions and requests if something goes wrong. An example would be to use UUID to make sure it is unique.
	
	Example Headers
	
	`Authorization: Bearer cn389ncoiwuencr`
	`Accept: application/json;charset=UTF-8`
	`X-DAS-TX: 123e4567-e89b-12d3-a456-426655440000`
	`X-DAS-CURRENCY: USD`
	`X-DAS-TZ: UTC+9`
	`X-DAS-LANG: en`

Money	Any monetary data in Dashur follows ISO standards when it comes to rounding, number of decimals etc **See all supported currencies and supported number of decimals in Appendix A** Dashur follows the following logic when processing incoming amounts 1. Remove trailing zeroes100.12000 becomes 100.12 2. Check if amount of decimals are same or less than the max amount of decimals allowed for the currency. Most currencies allow two decimals, as an example KRW and JPY don't allow any decimals See **https://en.wikipedia.org/wiki/ISO_4217#Active_codes** for a full list of allowed number of decimals. 3. If to many decimals are provided then reject the request **4.** If decimals are ok then process the request Dashur follows the following logic when sending currencies out

ITEM	DEFINITION
	1. Uses Round Half to Even algorithm (Bankers Rounding) when adding or subtracting totals https://en.wikipedia.org/wiki/Rounding#Round_half_to_even 2. Round to the requested currencies number of decimals. Most normally this will be two decimals but for currencies like KRW and JPY there will be no decimals. See **https://en.wikipedia.org/wiki/ISO_4217#Active_codes** for a full list of allowed number of decimals.

ITEM	DEFINITION
Success Response	All responses from the Dashur system are of type JSON. Most responses will come as a JSON Object with two properties meta and data. The meta field contains data such as requested currency, timezone, transaction id and the internal processing time of the request. If the request is paginated then there will also be a pagination object under the meta object which shows current page, page size, available pages etc Please note that for a paginated result totalPages and totalResults may be -1 if the system hasn't calculated them. If that's the case then the API caller can call the API while increasing the current "page" until the result is empty. **Example Non Paginated Response** `{ "meta": { "currency": "USD", "time_zone": "UTC", "transaction_id": "XXXX-XXX", "processing_time": -1 } "data": { "property": "value" } }`

ITEM	DEFINITION
Success Response (Continued)	**Example Paginated Response**

```
{
  “meta”: {
    "currency": "USD",
    "time_zone": "UTC",
    "transaction_id": "XXXX-XXX",
    "processing_time": -1,
    “pagination”: {
      “page”: 1,
      “pageSize”: 500,
      “totalPages”: 10,
      “totalResults”: 5000
  }
  “data”: {
    [
      {
        “property”: “value”
      }
     ]
  }
}
```

ITEM	DEFINITION
Timestamp	Dashur uses UTC offsets for timezones **See Appendix C for a full list of supported offsets** A timestamp follows these format 2015-11-01 -> yyyy-MM-dd 2015-11-01T17:12:45 -> yyyy-MM-ddTHH:mm:ss 2015-11-01T17:12:45 +0800 -> yyyy-MM-ddTHH:mm:ss ZZ 2015-11-01T17:12:45 UTC+0800 -> yyyy-MM-ddTHH:mm:ss 'UTC'ZZ, please note only UTC is accepted 2015-11-01T17:12:45.000 -> yyyy-MM-dd HH:mm:ss.SSS **Note that when passing as URL GET parameters make sure to URL encode the timestamp if it includes a space character or replace it with %20 which is the url encoded value of space** **Example** **Before: 2015-11-01 17:12:45 UTC+0800** **After: 2015-11-01%2017:12:45%20UTC+0800**

AUTHENTICATION

The Dashur API works by first calling the Auth service to login to the API. The response will then be an access_token and refresh_token. The access_token is valid for 1 hour and used to call the API endpoints. The refresh_token is valid for 3 hours and can be used to get a new access_token without providing the initial username and password. The use of the refresh_token is to minimize sending the username/password across the network.

When requesting the token you need to send Basic Authentication to the token endpoint. The authentication is the Auth Username and Auth Secret separated by : and base64 encoded.

Authenticate	
URL	/oauth/token
Method	POST

Example Create Basic Authentication

```
Auth username: testauthuser
Auth secret: testsecret
$ echo "testauthuser:testsecret" | base64
dGVzdGF1dGh1c2VyOnRlc3RzZWNyZXQK
```

Alternatively the site https://www.base64encode.org/ can be used to encode the token.

Example Request

```
POST /oauth/token
Authorization: Basic dGVzdGF1dGh1c2VyOnRlc3RzZWNyZXQK
Accept: application/json;charset=UTF-8
X-DAS-TX: 123e4567-e89b-12d3-a456-426655440000
X-DAS-CURRENCY: USD
X-DAS-TZ: UTC+9
X-DAS-LANG: en

grant_type=password&username=apiusername&password=apipassword
```

Example Response

```
{"access_token":"xxxxxxxxxxxxxxx","token_type":"bearer","refresh_token":"x
xxxxxxxx","expires_in":3599,"scope":"scope1:r,scope2:w","jti":"a2426760-
10e8-4ed3-95c9-a9ac991eaec6"}
```

The tokens in the response are JWT tokens and can be decoded at https://jwt.io/. The token includes data such as account id, username, scopes allowed etc.

Curl Example

```
curl -H "Content-Type: application/x-www-form-urlencoded " \
     -H "Authorization: Basic dGVzdGF1dGh1c2VyOnRlc3RzZWNyZXQK" \
     -H "X-DAS-TX-ID: tx_id_1234" \
     -H "X-DAS-TZ: UTC+8" \
     -H "X-DAS-LANG: en" \
     -H "X-DAS-CURRENCY: USD" \
     -X POST \
     -d 'grant_type=password&username=testapiuser&password=
testapipassword' \
     https://{{base url}}/oauth/token
```

Make sure to change base_url to the hostname provided and also switch out the usernames and secrets to the ones provided

CREATE MEMBER API

API calls to create new members

Create Member	
URL	/v1/account/member
Method	POST

Request

JSON Request	Format	Mandatory	Description
$.parent_id	64-bit Integer	Yes	The id of the parent account, can be the base account id provided unless sub company accounts are used
$.username	String (6-100)	Yes	The member's username. This username doesn't have to be unique in Dashur, instead the API will append a company tag (unique number for each Company) that makes the username unique. Example: username1 becomes username1:<<company_tag>>
$.password	String (6-100)	Yes	The password of the member
$.ext_ref	String (2-50)	No	Account external reference, this parameter can be used to set to the integrators internal id. If it is set, it will be provided back in transaction feeds and can also be used to generate launch urls, get accounts etc.

JSON Request	Format	Mandatory	Description
			It needs to be unique across the integrators tree
$.group_id	64-bit Integer	No	The group to put the member in. If not group_id is provided then the member will be put in the parents default group.

Response

JSON Response	Format	Description
$. data.meta	Object	The standard non paginated meta response. Including processing time, transaction_id, currency and timezone
$. data.data.id	64-bit Integer	The account id of the created member
$. data.my_path	String (2-100)	The path of the account in the Dashur tree structure. Example: <parents_parent id>.<parent_id>.<member_id>
$ data.user_id	64-bit Integer	The id of the connected user account. Each member account has a connected user account.
$ data.username	String (6-100)	The members username including the company tag appended at the end
$. data.user_status	Enum • ENABLED • DISABLED	The connected user accounts status in the system. It will always be ENABLED when creating a new member
$. data.parent_id	64-bit Integer	The id of the parent account
$. data.name	String (2-100)	The name of the user, this will be the same as the username passed in
$. data.ext_ref	String(2-50)	The account external reference provided

JSON Response	Format	Description
$. data.type	Enum • MEMBER	This will always be MEMBER. If using the account API then account type can also be COMPANY and SUBCOMPANY
$. data.currency_unit	String(3)	The currency for the member. This will be the same as the currency set for the parent account
$. data.status	Enum • ENABLED • DISABLED • SUSPENDED	This is the status for the account (not to be confused with the connected user account). It will always be ENABLED for a new member. ENABLED – Fully Active Account DISABLED – Locked out of the system SUSPENDED – Read only mode
$. data.test	Boolean	If the account is a test account. This property is for future use and is currently always false
$. data.version	64-bit Integer	The version of the account, the version is incremented each time the account is updated. If an update is attempted with an old version then the API will throw an error stating that there is a conflict during the update
$. data.created_by	64-bit Integer	The id of the user who created the entity
$. data.created	Timestamp	The time of creating the member
$. data.updated_by	64-bit Integer	The id of the user who update the member
$. data.updated	Timestamp	The time of updating the member

Example creating member

```
POST /v1/account/member
{
  "parent_id": 1234567,
  "username": "username1",
  "password": "password",
  "ext_ref": "user_ext_ref",
  "group_id": 1
}
```

Example response creating member


```
{
  "meta": {
    "currency": "USD",
    "time_zone": "UTC+08:00",
    "transaction_id": "tx_id_1234",
    "processing_time": 235
  },
  "data": {
    "id": 1111111,
    "my_path": "1234567, 1111111",
    "version": 1,
    "user_id": 12345,
    "username": "username1:1154",
    "user_status": "ENABLED",
    "parent_id": 1234567,
    "name": "username1",
    "ext_ref": "account_ext_ref",
    "type": "MEMBER",
    "currency_unit": "USD",
    "status": "ENABLED",
    "test": false,
    "created_by": 11111,
    "created": "2016-01-01 15:00:16.123 ",
    "updated_by": 11111,
    "updated": "2016-09-26 15:00:16.123"
  }
}
```

Curl Example

```
$ curl -H "Content-Type: application/json" \
     -H "Authorization: Bearer accesstokenXXXXX" \
     -H "X-DAS-TX-ID: tx_id_1234" \
     -H "X-DAS-TZ: UTC+8" \
     -H "X-DAS-LANG: en" \
     -H "X-DAS-CURRENCY: USD" \
     -X POST \
     -d '{"parent_id": 1234567,
"username":"username1","password":"userpassword1", "ext_ref":
"account_ext_ref", "group_id": 1}' \
https://{{base_url}}/v1/account/member
```

UPDATE MEMBER PASSWORD API

API Call to update a members password.

Update Password	
URL	/v1/account/member/password
Method	PUT

Request

JSON Request	Format	Mandatory	Description
$.account_id	64-bit Integer	No	The account_id of the member. Either account_id or ext_ref has to be provided
$.ext_ref	String (2-50)	No	The account external reference provided. Either account_id or ext_ref has to be provided
$.password	String (8-100)	Yes	The new password of the member

Response

JSON Response	Format	Description
$.meta	Object	The standard non paginated meta response. Including processing time, transaction_id, currency and timezone
$.data.id	64-bit Integer	The account id of the created member
$.data.my_path	String (2-100)	The path of the account in the Dashur tree structure. Example: <parents_parent id>.<parent_id>.<member_id>
$. data.user_id	64-bit Integer	The id of the connected user account. Each member account has a connected user account.
$. data.username	String (6-100)	The members username including the company tag appended at the end
$. data.user_status	Enum • ENABLED • DISABLED	The connected user accounts status in the system. It will always be ENABLED when creating a new member
$. data.parent_id	64-bit Integer	The id of the parent account
$. data.name	String (2-100)	The name of the user, this will be the same as the username passed in
$. data.ext_ref	String(2-50)	The account external reference provided

JSON Response	Format	Description
$. data.type	Enum • MEMBER	This will always be MEMBER. If using the account API then account type can also be COMPANY and SUBCOMPANY
$. data.currency_unit	String(3)	The currency for the member. This will be the same as the currency set for the parent account
$. data.status	Enum • ENABLED • DISABLED • SUSPENDED	This is the status for the account (not to be confused with the connected user account). It will always be ENABLED for a new member ENABLED – Fully Active Account DISABLED – Locked out of the system SUSPENDED – Read only mode
$. data.test	Boolean	If the account is a test account. This property is for future use and is currently always false
$. data.version	64-bit Integer	The version of the account, the version is incremented each time the account is updated. If an update is attempted with an old version then the API will throw an error stating that there is a conflict during the update
$. data.created_by	64-bit Integer	The id of the user who created the entity
$. data.created	Timestamp	The time of creating the member
$. data.updated_by	64-bit Integer	The id of the user who update the member
$. data.updated	Timestamp	The time of updating the member

Example updating password with account_id

```
POST /v1/account/member
{
  "account_id ": 1111111,
  "password”: “password1234”
}
```

Example response updating member password

```
{
  "meta": {
    "currency": "USD",
    "time_zone": "UTC+08:00",
    "transaction_id": "tx_id_1234",
    "processing_time": 235
  },
  "data": {
    "id": 1111111,
    "my_path": "1234567, 1111111",
    "version": 1,
    "user_id": 12345,
    "username": "username1:1154",
    "user_status": "ENABLED",
    "parent_id": 1234567,
    "name": "username1",
    "ext_ref": "account_ext_ref",
    "type": "MEMBER",
    "currency_unit": "USD",
    "status": "ENABLED",
    "test": false,
    "created_by": 11111,
    "created": "2016-01-01 15:00:16.123 ",
    "updated_by": 11111,
    "updated": "2016-09-26 15:00:16.123"
  }
}
```

Curl Example

```
curl -H "Content-Type: application/json" \
     -H "Authorization: Bearer accesstokenXXXXX" \
     -H "X-DAS-TX-ID: tx_id_1234" \
     -H "X-DAS-TZ: UTC+8" \
     -H "X-DAS-LANG: en" \
     -H "X-DAS-CURRENCY: USD" \
     -X PUT \
     -d '{"account_id": 1111111, "password":"password1234"}' \
https://{{base_url}}/v1/account/member/password
```

GET ACCOUNT BY ID/EXT_REF API

API call to get an account by id or ext_ref

Get Account	
URL	/v1/account/{account_id} /v1/account?ext_ref={ext_ref}
Method	GET

Request

Request	Format	Mandatory	Description
account_id	64-bit Integer	No	The account_id of the account. Either account_id or ext_ref has to be provided
ext_ref	String (2-50)	No	The account external reference provided. Either account_id or ext_ref has to be provided

Response

JSON Response	Format	Description
$.meta	Object	The standard non paginated meta response. Including processing time, transaction_id, currency and timezone
$.data.id	64-bit Integer	The account id of the created member
$.data.my_path	String (2-100)	The path of the account in the Dashur tree structure. Example: <parents_parent id>.<parent_id>.<member_id>
$. data.parent_id	64-bit Integer	The id of the parent account
$. data.name	String (2-100)	The name of the user, this will be the same as the username passed in
$. data.ext_ref	String(2-50)	The account external reference provided
$. data.type	Enum • MEMBER • SUB_COMPANY • COMPANY	This will always be MEMBER. If using the account API then account type can also be COMPANY and SUB_COMPANY
$. data.currency_unit	String(3)	The currency for the member. This will be the same as the currency set for the parent account
$. data.status	Enum • ENABLED • DISABLED • SUSPENDED	This is the status for the account (not to be confused with the connected user account).

JSON Response	Format	Description
		It will always be ENABLED for a new member ENABLED – Fully Active Account DISABLED – Locked out of the system SUSPENDED – Read only mode
$. data.test	Boolean	If the account is a test account. This property is for future use and is currently always false
$. data.version	64-bit Integer	The version of the account, the version is incremented each time the account is updated. If an update is attempted with an old version then the API will throw an error stating that there is a conflict during the update
$. data.created_by	64-bit Integer	The id of the user who created the account
$. data.created	Timestamp	The time the account was created
$. data.updated_by	64-bit Integer	The id of the user who update the account last
$.data.updated	Timestamp	The last time the account was updated

Example get account by id / by ext_ref

```
GET /v1/account/1111111
GET /v1/account?ext_ref=account_ext_ref
```

Example response updating member password

```
{
  "meta": {
    "currency": "USD",
    "time_zone": "UTC+08:00",
    "transaction_id": "tx_id_1234",
    "processing_time": 33
  },
  "data": {
    "id": 1111111,
    "my_path": "1234567,1111111",
    "version": 1,
    "parent_id": 1234567,
    "name": "username1",
    "ext_ref": "account_ext_ref",
    "type": "MEMBER",
    "currency_unit": "USD",
    "status": "ENABLED",
    "test": false,
    "created_by": 19452,
    "created": "2016-01-01 15:53:16.872",
    "updated_by": 19452,
    "updated": "2016-01-01 15:53:16.872"
  }
}
```

Curl Example

```
curl -H "Content-Type: application/json" \
     -H "Authorization: Bearer accesstokenXXXXX" \
     -H "X-DAS-TX-ID: tx_id_1234" \
     -H "X-DAS-TZ: UTC+8" \
     -H "X-DAS-LANG: en" \
     -H "X-DAS-CURRENCY: USD" \
     -X GET \
https://{{base_url}}/v1/account/1111111
```

LIST CHILD ACCOUNTS

API call to list all the children of an account

List Child Accounts	
URL	/v1/account/{account_id}/children
Method	GET

Request

Request	Format	Mandatory	Description
account_id	64-bit Integer	Yes	The account_id of the account whose children should be listed. This param is in the path
page	32-bit Integer	No	The page to list, will default to 1
page_size	32-bit Integer	No	The page size requested. The default is 50
desc	Boolean	No	If the listing should be done descending. Default false

Response

JSON Response	Format	Description
$.meta	Object	The standard paginated meta response. Including processing time, transaction_id, currency and timezone
$.data[].id	64-bit Integer	The account id of the created member
$.data[].my_path	String (2-100)	The path of the account in the Dashur tree structure. Example: <parents_parent id>.<parent_id>.<member_id>
$. data[].parent_id	64-bit Integer	The id of the parent account
$. data[].name	String (2-100)	The name of the user, this will be the same as the username passed in
$. data[].ext_ref	String(2-50)	The account external reference provided
$. data[].type	Enum • MEMBER • SUB_COMPANY • COMPANY	This will always be MEMBER. If using the account API then account type can also be COMPANY and SUB_COMPANY
$.data[].currency_unit	String(3)	The currency for the member. This will be the same as the currency set for the parent account

JSON Response	Format	Description
$. data[].status	Enum • ENABLED • DISABLED • SUSPENDED	This is the status for the account (not to be confused with the connected user account). It will always be ENABLED for a new member
$. data[].test	Boolean	If the account is a test account. This property is for future use and is currently always false
$. data[].version	64-bit Integer	The version of the account, the version is incremented each time the account is updated. If an update is attempted with an old version then the API will throw an error stating that there is a conflict during the update
$. data[].created_by	64-bit Integer	The id of the user who created the account
$. data[].created	Timestamp	The time the account was created
$. data[].updated_by	64-bit Integer	The id of the user who update the account last
$.data[].updated	Timestamp	The last time the account was updated

Example get children of an account

```
GET /v1/account/1111111/children
```

Example response of get children of account

```
{
  "meta": {
    "currency": "USD",
    "time_zone": "UTC+08:00",
    "transaction_id": "tx_id_1234",
    "processing_time": 51,
    "pagination": {
      "page": 1,
      "page_size": 50,
      "total_pages": 160,
      "total_results": 7954
    }
  }
  "data": [{
    "id": 1111111,
    "my_path": "1234567,1111111",
    "version": 1,
    "parent_id": 1234567,
    "name": "username1",
    "ext_ref": "account_ext_ref",
    "type": "MEMBER",
    "currency_unit": "USD",
    "status": "ENABLED",
    "test": false,
    "created_by": 19452,
    "created": "2016-01-01 15:53:16.872",
    "updated_by": 19452,
    "updated": "2016-01-01 15:53:16.872"
  }]
}
```

Curl Example

```
curl -H "Content-Type: application/json" \
     -H "Authorization: Bearer accesstokenXXXXX" \
     -H "X-DAS-TX-ID: tx_id_1234" \
     -H "X-DAS-TZ: UTC+8" \
     -H "X-DAS-LANG: en" \
     -H "X-DAS-CURRENCY: USD" \
     -X GET \
https://{{base_url}}/v1/account/1111111/children?page=1&page_size=100
```

CREATE TRANSACTION

API to debit/credit credits. When using the credit light model there the TRANSFER category for a transaction is used. This type of transaction transfers credits from an parent account to a child account or vice versa. Therefor for each transaction created two transactions will be returned. One for the account the transaction was executed against and one for the parent account.

Create Transaction	
URL	/v1/transaction
Method	POST

Request

Multiple transactions against the same account can be sent at the same time. This can be used to split up different transactions in smaller parts. Such as CC Fee and the amount to be credited or to execute transactions against the credit balance at the same time. When multiple transactions are sent they will all be committed atomically.

JSON Request	Format	Mandatory	Description
$.[].account_id	64-bit Integer	No	The account_id of the account the transaction will be created for. **Either account_id or account_ext_ref has to be supplied**
$.data.[].account_ext_ref	String (2-50)	No	The external reference of the account the transaction will be created for. **Either account_id or account_ext_ref has to be supplied**
$.[].external_ref	String(0-50)	No	An external reference of a transaction which will be supplied back in the feed api
$.[].amount	Money	Yes	The amount to credit or debit to the wallet
$.[].type	Enum • CREDIT • DEBIT	Yes	If this is a credit or debit transaction. If credit then then amount will be added to the current balance and if debit then the amount will be subtracted

JSON Request	Format	Mandatory	Description
$.[].balance_type	Enum • CREDIT_BALANCE	Yes	The balance to execute the transaction against.
$.[].category	Enum • TRANSFER	Yes	The transaction category. Currently credit light wallet can only support TRANSFER transactions.
$.[].meta_data	JSON Map(1000)	No	The meta data object is a JSON Map that can contain any fields that the caller wants to tag the transaction with. This data follows the transaction through the system and will be present when requesting the transaction at a later date. Example of usage would be to tag the transaction with promotion ids, session ids or any other data needed for later processing If a key called “description” is provided then it will be used in the Dashur UI to show an explanation of the Transaction

Response

Please note that for every transaction sent there will be two transactions returned. One for the parent account and one for the account the transaction is for. If the transaction is a CREDIT then the second transaction that is returned will be a DEBIT from the parent account

JSON Response	Format	Description
$.meta	Object	The standard non paginated meta response. Including processing time, transaction_id, currency and timezone
$.data.[].id	64-bit Integer	The id of the created transaction
$.data.[].parent_transaction_id	64-bit Integer	The id of any linked transaction. If there is no linked transaction the the parent_id is the same as the id of the transaction
$.data.[].account_ext_ref	String (2-50)	The account external reference, either
$.data.[].application_id	64-bit Integer	The application that created the transaction
$.data.[].currency_unit	String(3)	The currency code used for the transaction
$.data.[].transaction_time	Timestamp	The time the transaction was created
$.data.[].balance	Money	The new balance of the wallet that the transaction was executed against

JSON Response	Format	Description
$.data.[].external_ref	String(0-50)	The external id of the transaction.
$.data.[].session	String(0-32)	The login session that the transaction was created in.
$. data.[].ip	String	The IP of the API user who created the transaction
$. data.[]. wallet_code	String(2-20)	The wallet the transaction was executed against
$. data.[]. category	Enum • Transfer	The transaction category.
$. data.[].type	Enum • CREDIT • DEBIT	If the wallet was debited or credited
$. data.[]. balance_type	Enum • CREDIT_BALANCE	The balance that was updated. Currently for a credit light wallet there is only one balance which is the Credit Balance.
$. data.[]. amount	Money	The amount of the transaction
$. data.[].created_by	64-bit Integer	The id of the user who created the account
$. data.[].created	Timestamp	The time the account was created
$.data.[].meta_data	JSON Map(1000)	The meta data of the transaction

Example request create a transaction

```
POST /v1/transaction

[{
    "account_id": 11222,
    "category": "TRANSFER",
    "type": "CREDIT",
    "balance_type": "CREDIT_BALANCE",
    "amount": "10000.10",
    "external_ref": "ext1234"
    "meta_data": {
      "description": "Credit account",
      "mypromokey": "promokey1"
    }
}]
```

Example response creating a transaction

```
{
  "meta": {
    "currency": "USD",
    "time_zone": "UTC+08:00",
    "transaction_id": "tx_id_1234",
    "processing_time": 216
  },
  "data": [
    {
      "id": 111111,
      "parent_transaction_id": 11111,
      "account_id": 11222,
      "account_ext_ref": "account_ext_ref",
      "application_id": 1234,
      "currency_unit": "USD",
      "transaction_time": "2016-01-01 22:02:44.631",
      "balance": 10000.1,
      "created_by": 112233,
      "created": "2016-01-01 22:02:44.631",
      "external_ref": "ext1234"
      "session": "19751-19452-19",
      "ip": "172.16.1.1",
      "wallet_code": "CREDIT_LIGHT1",
      "category": "TRANSFER",
      "balance_type": "CREDIT_BALANCE",
      "type": "CREDIT",
      "amount": 10000.1,
      "meta_data": {
        "description": "Credit account",
        "mypromokey": "promokey1",
      }
    },
    {
      “id”: 121212
      "type": "DEBIT",
      "account_id": 111111 //parent of 11222
      ...
    }

  ]
}
```

Curl Example

```
curl -H "Content-Type: application/json" \
     -H "Authorization: Bearer accesstokenXXXXX" \
     -H "X-DAS-TX-ID: tx_id_1234" \
     -H "X-DAS-TZ: UTC+8" \
     -H "X-DAS-LANG: en" \
     -H "X-DAS-CURRENCY: USD" \
     -X POST \
     -d '[{"account_id": 11222, "category": "TRANSFER", "type": "CREDIT",
“balance_type”: “\}”, "amount": "10000.10", "external_ref": "ext123",
"meta_data": {"description": "Credit account"} }]' \
     http://{{base_url}}/v1/transaction
```

VERIFY TRANSACTION

API to verify if a transaction has been committed. This can be used if as an example there is a network error while performing a transaction against the member's wallet. This API can be called instantly after a failure and will respond if the transaction was committed up until one hour later. If there is a need to verify a transaction more than 1 hour after it was executed then the transaction feed API needs to be used.

Verify Transaction	
URL	/v1/transaction?ext_ref=?&account_id=?
Method	GET

Request

Request	Format	Mandatory	Description
account_id	64-bit Integer	No	The account_id of the account the transaction will be created for. **Either account_id or account_ext_ref has to be supplied**
account_ext_ref	String(2-50)	No	The external reference of the account the transaction will be created for. **Either account_id or account_ext_ref has to be supplied**
ext_ref	String (2-50)	Yes	The transactions external reference. This should be the value

Request	Format	Mandatory	Description
			that was used when executing the initial TX

Response

JSON Response	Format	Description
$.meta	Object	The standard non paginated meta response. Including processing time, transaction_id, currency and timezone
$.data.[].id	64-bit Integer	The id of the created transaction
$.data.[].parent_transaction_id	64-bit Integer	The id of any linked transaction. If there is no linked transaction the the parent_id is the same as the id of the transaction
$.data.[].account_ext_ref	String (2-50)	The account external reference
$.data.[].application_id	64-bit Integer	The application that created the transaction
$.data.[].currency_unit	String(3)	The currency code used for the transaction
$.data.[].transaction_time	Timestamp	The time the transaction was created

JSON Response	Format	Description
$.data.[].balance	Money	The new balance of the wallet that the transaction was executed against
$.data.[].external_ref	String(0-50)	The external reference used to create the transaction
$.data.[].session	String(0-32)	The login session that the transaction was created in.
$. data.[].ip	String	The IP of the API user who created the transaction
$. data.[]. wallet_code	String(2-20)	The wallet the transaction was executed against
$. data.[]. category	Enum • **TRANSFER**	The transaction category.
$. data.[].type	Enum • CREDIT • DEBIT	If the wallet was debited or credited
$. data.[]. balance_type	Enum • CREDIT_BALANCE	The balance that was updated. Currently for a credit light wallet there is only one balance which is the Credit Balance.
$. data.[]. amount	Money	The amount of the transaction
$. data.[].created_by	64-bit Integer	The id of the user who created the account
$. data.[].created	Timestamp	The time the account was created

JSON Response	Format	Description
$.data.[].meta_data	JSON Map(1000)	The meta data of the transaction

Example verify a transaction

```
GET /v1/transaction?account_id=11222&ext_ref=ext1234
```

Example response verifying a transaction


```
{
  "meta": {
    "currency": "USD",
    "time_zone": "UTC+08:00",
    "transaction_id": "tx_id_1234",
    "processing_time": 216
  },
  "data": [
    {
      "id": 111111,
      "parent_transaction_id": 11111,
      "account_id": 11222,
      "account_ext_ref": "account_ext_ref",
      "application_id": 1234,
      "currency_unit": "USD",
      "transaction_time": "2016-01-01 22:02:44.631",
      "balance": 10000.1,
      "created_by": 112233,
      "created": "2016-01-01 22:02:44.631",
      "session": "19751-19452-19",
      "ip": "172.16.1.1",
      "wallet_code": "CREDIT_LIGHT1",
      "category": "TRANSFER",
      "balance_type": "CREDIT_BALANCE",
      "type": "CREDIT",
      "amount": 10000.1,
      "meta_data": {
        "description": "Credit account",
        "mypromokey": "promokey1",
      }
    }
  ]
}
```

Curl Example

```
curl -H "Content-Type: application/json" \
     -H "Authorization: Bearer accesstokenXXXXX" \
     -H "X-DAS-TX-ID: tx_id_1234" \
     -H "X-DAS-TZ: UTC+8" \
     -H "X-DAS-LANG: en" \
     -H "X-DAS-CURRENCY: USD" \
     -X GET
'https://{{base_url}}/v1/transaction?account_id=28399&ext_ref=ext1234'
```

GET WALLET / BALANCE

API to get the wallet(s) of the member via account id or account ext_ref. Normally a member will only have one wallet but depending on the company configuration one member can have multiple wallets.

Get Balance	
URL	/v1/wallet?account_ext_ref=?&account_id=?&code=?
Method	GET

Request

Request	Format	Mandatory	Description
account_id	64-bit Integer	No	The account_id of the account the wallet will be retreived for. **Either account_id or account_ext_ref has to be supplied**
account_ext_ref	String(2-50)	No	The external reference of the account the wallet will be retreived for. **Either account_id or account_ext_ref has to be supplied**
code	String (2-20)	Yes	The code for the wallet if one specific wallet is being retrieved. The codes available can be retrieved via your account manager

Response

JSON Response	Format	Description
$.meta	Object	The standard non paginated meta response. Including processing time, transaction_id, currency and timezone
$.data.[].id	64-bit Integer	The id of the created transaction
$.data.[].type	Enum • **CREDIT**	The type of wallet.
$.data.[].version	64-bit Integer	The current version saved in the system. Each time the wallet is updated the version will be incremented
$.data.[]. account_id	64-bit Integer	The account id the wallet belongs to
$.data.[].code	String(2-20)	The code of the wallet, this is a ID the connects wallets across the tree of accounts, sub accounts and members. Normally the value will be "CREDIT_LIGHT1"
$.data.[].name_lang	Json Map	The name of the wallet in different languages (if set)
$.data.[].credit_balance	Money	The credit balance of the wallet
$.data.[].test	Boolean	If the wallet is a test wallet or not Currently will always be false
$. data[].created_by	64-bit Integer	The id of the user who created the wallet
$. data[].created	Timestamp	The time the wallet was created

JSON Response	Format	Description
$. data[].updated_by	64-bit Integer	The id of the user who update the wallet last
$.data[].updated	Timestamp	The last time the wallet was updated

Example request wallets

```
GET /v1/wallet?account_id=11222

GET /v1/wallet?account_ext_ref=XYZ1234
```

Example response getting wallets


```
{
  "meta": {
    "currency": "USD",
    "time_zone": "UTC+08:00",
    "transaction_id": "tx_id_1234",
    "processing_time": 202
  },
  "data": [
    {
      "type": "CREDIT_LIGHT",
      "id": 42799,
      "version": 1,
      "account_id": 28399,
      "code": "CREDIT_LIGHT1",
      "name": "Wallet Credit LIGHT",
      "test": false,
      "created_by": 19452,
      "created": "2016-09-26 15:53:16.872",
      "updated_by": 19452,
      "updated": "2016-09-26 15:53:16.872",
      "currency_unit": "USD",
      "credit_balance": 20000.2
    }
  ]
}
```

Curl Example

```
curl -H "Content-Type: application/json" \
     -H "Authorization: Bearer accesstokenXXXXX" \
     -H "X-DAS-TX-ID: tx_id_1234" \
     -H "X-DAS-TZ: UTC+8" \
     -H "X-DAS-LANG: en" \
     -H "X-DAS-CURRENCY: USD" \
     -X GET 'https://{{base_url}}/v1/wallet?account_id=28399'
```

LAUNCH ITEM API

API to generate a redirect link that can be used to redirect a member. The API will make sure the member gets logged in, generate the token, construct the url and send the url back in the response.

Launch Item	
URL	/v1/launcher/item
Method	POST

Request

JSON Request	Format	Mandatory	Description
$.account_id	64-bit Integer	No	The account_id to generated the link for. Either account_id or ext_ref or demo has to be provided
$.ext_ref	String (2-50)	No	The account external reference to generate the link for. Either account_id or ext_ref or demo has to be provided
$.demo	**Boolean**	**No**	When demo is set to true, the launch url generated will launch the game in demo mode. **Demo mode enabled for Flash and HTML5 RNG Slot Game.**
$.item_id	64-bit Int	Yes	The item to generate the link for.

JSON Request	Format	Mandatory	Description
$.app_id	**64-bit Int**	**Yes**	**The application to generate the link for**
$.login_context	JSON Map(1000)	No	The login context is used to enhance the reporting of transaction data across Transaction Feeds and Reporting Services.
$.login_context.promotion_key	String(0-32)	No	Any promotion the member is currently part of
$.login_context.lang	String(0-5)	No	The language of the members. This will be used for selecting the correct language in the launch urls. See Appendinx A for valid values.
$.login_context.country_id	String(0-2)	No	The country the member belongs to using ISO country codes (FR, US, CN) etc.
$.login_context.session_key	String(0-32)	No	The current session id of the member in the external system.
$.login_context.ip	String(0-45)	No	The IP of the member. This is used for IP restrictions and reporting
$.login_context.user_agent	String(0-1024)	No	The current browser user agent the Member is using. This allows reporting on device types, os versions etc

JSON Request	Format	Mandatory	Description
$.login_context.meta_data	JSON Map(1000)	No	The meta data object is a JSON Map that can contain any fields that the caller wants to tag the login context with. This data will then be added to any transaction and will follow through into transaction feeds
$.conf_params	JSON Map(1000)	No	The meta data object is a JSON Map that can contain any fields that the caller wants to tag the conf params with.
$.conf_params.lobby_url	String(255)	No	Define game-lobby page for HTML5 RNG slots game launch. This will enable the related button within the game, and perform the redirection when the button is clicked.
$.conf_params.banking_url	String(255)	No	Define banking page for HTML5 RNG slots game launch. This will enable the related button within the game, and perform the redirection when the button is clicked.
$.conf_params.logout_url	String(255)	No	Define default logout page for HTML5 Diamond Live Dealer game launch. Player will be redirect to this page once perform logout.

JSON Request	Format	Mandatory	Description
$.conf_params.failed_url	String(255)	No	Define failed page for HTML5 Diamond Live Dealer game launch. Player will be redirect to this page once perform failed.
$.conf_params.titanium	Enum • default • embedded • select	No	This is required when generate Live Dealer Titanium URL.

Response

JSON Response	Format	Description
$.meta	Object	The standard non paginated meta response. Including processing time, transaction_id, currency and timezone
$.data	String(0-2048)	The generated url

Example request to launch item

```
POST /v1/launcher/item

{
  "account_id": 12344,
  "item_id": 1111,
  "app_id": 11112,
  "login_context": {
    "ip": "192.168.1.100",
    "session_key": "xyz",
    "user_agent": "Mozilla\/5.0 (Windows NT 6.1; WOW64; rv:40.0"
  }
}
```

Example response launching an item

```
{
  "meta": {
    "currency": "USD",
    "time_zone": "UTC+08:00",
    "transaction_id": "tx_id_1234",
    "processing_time": 843
  },
  "data": "https://redirec.com/redirect_url.."
}
```

Curl Example

```
curl -H "Content-Type: application/json" \
     -H "Authorization: Bearer accesstokenXXXXX" \
     -H "X-DAS-TX-ID: tx_id_1234" \
     -H "X-DAS-TZ: UTC+8" \
     -H "X-DAS-LANG: en" \
     -H "X-DAS-CURRENCY: USD" \
     -X POST \
     -d '{"account_id": 12344, "item_id": 1111, "app_id": 11112,
"login_context": { "ip": "192.168.1.100", "session_key": "xyz",
"user_agent": "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100101
Firefox/40.1"}}' \
     https://{{base url}}/v1/launcher/item
```

LAUNCH TRANSACTION DETAIL API

API to generate a link to any transaction detail pages an Application provides.

Launch Transaction	
URL	/v1/launcher/tx/{transaction_id}?lang={lang}
Method	GET

Request

Request	Format	Mandatory	Description
Transaction_id	64-bit Integer	Yes	The Dashur transaction id to get the link for
lang	String(0-5)	No	The language to use for the transaction detail page. Will default to English if no code is provided or a non-supported code is provided.

Response

JSON Response	Format	Description
$.meta	Object	The standard non paginated meta response. Including processing time, transaction_id, currency and timezone
$.data	String(0-2048)	The url to the transaction detail page

Example request to launch transaction detail

```
GET /v1/launcher/tx/{12345}?lang=en
```

Example response launching transaction detail

```
{
  "meta": {
    "currency": "USD",
    "time_zone": "UTC+08:00",
    "transaction_id": "tx_id_1234",
    "processing_time": 843
  },
  "data": "https://redirec.com/redirect_url.."
}
```

Curl Example

```
curl -H "Content-Type: application/json" \
     -H "Authorization: Bearer accesstokenXXXXX" \
     -H "X-DAS-TX-ID: tx_id_1234" \
     -H "X-DAS-TZ: UTC+8" \
     -H "X-DAS-LANG: en" \
     -H "X-DAS-CURRENCY: USD" \
     -X GET \
     http://{{base_url}}/v1/launcher/transaction/12345?lang=en
```

TRANSACTION FEED

API to get a complete log of transactions

The Transaction feed API returns a list of transactions base on the various query parameters. The returned transactions are sorted by transaction time in descending order. There are two query patterns:

1. query by external_ref, returns all transactions that match the external ref
2. query by company_id, start_time, end_time, all parameters are mandatory. This will return all the transaction for members below this company

For query 2, all matched transactions are returned according to page size and page.

TX Feed	
URL	/v1/feed/transaction
Method	GET

Request by Ext Ref

Request	Format	Mandatory	Description
external_ref	String (0-50)	Yes	The external reference of the transaction.

Request by Company Id

Request	Format	Mandatory	Description
company_id	64-bit Integer	Yes	The account_id of the company to list transactions for. All Member transactions will be included
start_time	Timestamp	Yes	The starting time to list transactions from 2015-11-01 -> yyyy-MM-dd 2015-11-01T17:12:45 -> yyyy-MM-ddTHH:mm:ss 2015-11-01T17:12:45 +0800 -> yyyy-MM-ddTHH:mm:ss ZZ 2015-11-01T17:12:45 UTC+0800 -> yyyy-MM-ddTHH:mm:ss 'UTC'ZZ, please note only UTC is accepted 2015-11-01 17:12:45.000 -> yyyy-MM-dd HH:mm:ss.SSS
end_time	Timestamp	Yes	The end time to list transactions to Timestamp format is same as for start_date
include_transfers	**Boolean**	**No**	**To include TRANSFER** **tx in the response.** **Default value is true**
include_end_round	**Boolean**	**No**	**To include the END_ROUND transaction**

Request	Format	Mandatory	Description
			type which indicates a round has closed **Default value is false**
page_size	32-bit Integer	No	The size of the page to list

Response

JSON Response	Format	Description
$.meta	Object	The standard non paginated meta response. Including processing time, transaction_id, currency and timezone
$.data.[].id	64-bit Integer	The id of the created transaction
$.data.[].parent_transaction_id	64-bit Integer	The id of any linked transaction. If there is no linked transaction the the parent_id is the same as the id of the transaction
$.data.[].account_ext_ref	String (2-50)	The account external reference
$.data.[].application_id	64-bit Integer	The application that created the transaction
$.data.[].currency_unit	String(3)	The currency code used for the transaction
$.data.[].transaction_time	Timestamp	The time the transaction was created
$.data.[].balance	Money	The new balance of the wallet that the transaction was executed against
$.data.[].external_ref	String(0-50)	The external reference used to create the transaction
$.data.[].session	String(0-32)	The login session that the transaction was created in.

JSON Response	Format	Description
$. data.[].ip	String	The IP of the API user who created the transaction
$. data.[]. wallet_code	String(2-20)	The wallet the transaction was executed against
$. data.[]. category	Enum • **TRANSFER** • WAGER • PAYOUT • END_ROUND • REFUND	The transaction category.
$. data.[].type	Enum • CREDIT • DEBIT	If the wallet was debited or credited
$. data.[]. balance_type	Enum • CREDIT_BALANCE	The balance that was updated.
$. data.[]. amount	Money	The amount of the transaction
$. data.[].created_by	64-bit Integer	The id of the user who created the account
$. data.[].created	Timestamp	The time the account was created
$.data.[].meta_data	JSON Map(1000)	The meta data of the transaction

Example request to to get a feed

```
GET /v1/feed/transaction?company_id=111111&start_time=2016-01-
26&end_time=2016-01-28

GET /v1/feed/transaction?ext_ref=ext123
```

Example response from transaction feed


```
{
  "meta": {
    "currency": "USD",
    "time_zone": "UTC+08:00",
    "transaction_id": "tx_id_1234",
    "processing_time": 312,
    "pagination": {
      "page": 1,
      "page_size": 1,
      "total_pages": 1,
      "total_results": 1
    }
  },
  "data": [
    {
      "id": 128001,
      "parent_transaction_id": 128001,
      "account_id": 1111111,
      "account_ext_ref": "account_ext_ref",
      "application_id": 1560,
      "currency_unit": "USD",
      "transaction_time": "2016-01-01 13:21:33.591",
      "balance": 20000.2,
      "created_by": 19452,
      "created": "2016-01-01 13:21:33.591",
      "session": "19751-19452-22",
      "ip": "172.16.1.1",
      "wallet_code": "CREDIT_LIGHT1",
      "external_ref": "ext1234",
      "category": "TRANSFER",
      "sub_category": "",
      "balance_type": "LIGHT_BALANCE",
      "type": "CREDIT",
      "amount": 10000.1,
      "meta_data": {
        "description": "Credit account"
      }
    }
  ]
}
```

Curl Example

```
curl -H "Content-Type: application/json" \
     -H "Authorization: Bearer accesstokenXXXXX" \
     -H "X-DAS-TX-ID: tx_id_1234" \
     -H "X-DAS-TZ: UTC+8" \
     -H "X-DAS-LANG: en" \
     -H "X-DAS-CURRENCY: USD" \
     -X GET \
'https://{{base_url}}/v1/feed/transaction?company_id=111111&start_time=201
6-01-26&end_time=2016-01-28'
```

APPENDIX A SUPPORTED LOCALES

Language	Country	Locale ID
Albanian	Albania	sq_AL
Arabic	Algeria	ar_DZ
Arabic	Bahrain	ar_BH
Arabic	Egypt	ar_EG
Arabic	Iraq	ar_IQ
Arabic	Jordan	ar_JO
Arabic	Kuwait	ar_KW
Arabic	Lebanon	ar_LB
Arabic	Libya	ar_LY
Arabic	Morocco	ar_MA
Arabic	Oman	ar_OM
Arabic	Qatar	ar_QA
Arabic	Saudi Arabia	ar_SA

Language	Country	Locale ID
Arabic	Sudan	ar_SD
Arabic	Syria	ar_SY
Arabic	Tunisia	ar_TN
Arabic	United Arab Emirates	ar_AE
Arabic	Yemen	ar_YE
Belarusian	Belarus	be_BY
Bulgarian	Bulgaria	bg_BG
Catalan	Spain	ca_ES
Chinese (Simplified)	China	zh_CN
Chinese (Simplified)	Singapore	zh_SG
Chinese (Traditional)	Hong Kong	zh_HK
Chinese (Traditional)	Taiwan	zh_TW
Croatian	Croatia	hr_HR

Language	Country	Locale ID
Czech	Czech Republic	cs_CZ
Danish	Denmark	da_DK
Dutch	Belgium	nl_BE
Dutch	Netherlands	nl_NL
English	Australia	en_AU
English	Canada	en_CA
English	India	en_IN
English	Ireland	en_IE
English	Malta	en_MT
English	New Zealand	en_NZ
English	Philippines	en_PH
English	Singapore	en_SG
English	South Africa	en_ZA

Language	Country	Locale ID
English	United Kingdom	en_GB
English	United States	en_US
Estonian	Estonia	et_EE
Finnish	Finland	fi_FI
French	Belgium	fr_BE
French	Canada	fr_CA
French	France	fr_FR
French	Luxembourg	fr_LU
French	Switzerland	fr_CH
German	Austria	de_AT
German	Germany	de_DE
German	Luxembourg	de_LU
German	Switzerland	de_CH

Language	Country	Locale ID
Greek	Cyprus	el_CY(*)
Greek	Greece	el_GR
Hebrew	Israel	iw_IL
Hindi	India	hi_IN
Hungarian	Hungary	hu_HU
Icelandic	Iceland	is_IS
Indonesian	Indonesia	in_ID
Irish	Ireland	ga_IE
Italian	Italy	it_IT
Italian	Switzerland	it_CH
Japanese (Gregorian calendar)	Japan	ja_JP
Japanese (Imperial calendar)	Japan	ja_JP_JP

Language	Country	Locale ID
Korean	South Korea	ko_KR
Latvian	Latvia	lv_LV
Lithuanian	Lithuania	lt_LT
Macedonian	Macedonia	mk_MK
Malay	Malaysia	ms_MY
Maltese	Malta	mt_MT
Norwegian (Bokmål)	Norway	no_NO
Norwegian (Nynorsk)	Norway	no_NO_NY
Polish	Poland	pl_PL
Portuguese	Brazil	pt_BR(*)
Portuguese	Portugal	pt_PT(*)
Romanian	Romania	ro_RO
Russian	Russia	ru_RU

Language	Country	Locale ID
Serbian (Cyrillic)	Bosnia and Herzegovina	sr_BA
Serbian (Cyrillic)	Montenegro	sr_ME
Serbian (Cyrillic)	Serbia	sr_RS
Serbian (Latin)	Bosnia and Herzegovina	sr_Latn_BA
Serbian (Latin)	Montenegro	sr_Latn_ME
Serbian (Latin)	Serbia	sr_Latn_RS
Slovak	Slovakia	sk_SK
Slovenian	Slovenia	sl_SI
Spanish	Argentina	es_AR
Spanish	Bolivia	es_BO
Spanish	Chile	es_CL
Spanish	Colombia	es_CO
Spanish	Costa Rica	es_CR

Language	Country	Locale ID
Spanish	Dominican Republic	es_DO
Spanish	Ecuador	es_EC
Spanish	El Salvador	es_SV
Spanish	Guatemala	es_GT
Spanish	Honduras	es_HN
Spanish	Mexico	es_MX
Spanish	Nicaragua	es_NI
Spanish	Panama	es_PA
Spanish	Paraguay	es_PY
Spanish	Peru	es_PE
Spanish	Puerto Rico	es_PR
Spanish	Spain	es_ES
Spanish	United States	es_US

Language	Country	Locale ID
Spanish	Uruguay	es_UY
Spanish	Venezuela	es_VE
Swedish	Sweden	sv_SE
Thai (Western digits)	Thailand	th_TH
Thai (Thai digits)	Thailand	th_TH_TH
Turkish	Turkey	tr_TR
Ukrainian	Ukraine	uk_UA
Vietnamese	Vietnam	vi_VN

APPENDIX B SUPPORTED CURRENCIES

Currency	Decimals
AED	2
AFN	2
ALL	2
AMD	2
ANG	2
AOA	2
ARS	2
AUD	2
AWG	2
AZN	2
BAM	2
BBD	2
BDT	2
BGN	2
BHD	3
BIF	0

Currency	Decimals
BMD	2
BND	2
BOB	2
BRL	2
BSD	2
BTN	2
BWP	2
BYR	0
BZD	2
CAD	2
CDF	2
CHF	2
CLF	0
CLP	0
CNY	2
COP	2
CRC	2
CUC	2

Currency	Decimals
CUP	2
CVE	2
CZK	2
DJF	0
DKK	2
DOP	2
DZD	2
EEK	2
EGP	2
ERN	2
ETB	2
EUR	2
FJD	2
FKP	2
GBP	2
GEL	2
GHS	2
GIP	2

Currency	Decimals
GMD	2
GNF	0
GTQ	2
GYD	2
HKD	2
HNL	2
HRK	2
HTG	2
HUF	2
IDR	2
ILS	2
INR	2
IQD	3
IRR	2
ISK	0
JMD	2
JOD	3
JPY	0

Currency	Decimals
KES	2
KGS	2
KHR	2
KMF	0
KPW	2
KRW	0
KWD	3
KYD	2
KZT	2
LAK	2
LBP	2
LKR	2
LRD	2
LSL	2
LTL	2
LVL	2
LYD	3
MAD	2

Currency	Decimals
MDL	2
MGA	2
MKD	2
MMK	2
MNT	2
MOP	2
MRO	2
MTL	2
MUR	2
MVR	2
MWK	2
MXN	2
MYR	2
MZN	2
NAD	2
NGN	2
NIO	2
NOK	2

Currency	Decimals
NPR	2
NZD	2
OMR	3
PAB	2
PEN	2
PGK	2
PKR	2
PLN	2
PYG	0
QAR	2
RON	2
RSD	2
RUB	2
RWF	0
SAR	2
SBD	2
SCR	2
SDG	2

Currency	Decimals
SEK	2
SHP	2
SLL	2
SOS	2
SRD	2
STD	2
SVC	2
SYP	2
SZL	2
THB	2
TJS	2
TMT	2
TND	3
TOP	2
TRY	2
TTD	2
TZS	2
UAH	2

Currency	Decimals
UGX	0
USD	2
UYU	2
UZS	2
VEF	2
VND	0
VUV	0
WST	2
XAF	0
XCD	2
XOF	0
XPF	0
YER	2
ZAR	2
ZMK	2
ZMW	2
ZWL	2

APPENDIX C SUPPORTED TIMEZONES

A list of countries and their respective UTC offsets can be found at https://en.wikipedia.org/wiki/List_of_UTC_time_offsets

Dashur UTC Offset Codes
UTC-12
UTC-11
UTC-10
UTC-9
UTC-8
UTC-7
UTC-6
UTC-5
UTC-4
UTC-3
UTC-2
UTC-1
UTC
UTC+1
UTC+2

Dashur UTC Offset Codes
UTC+3
UTC+4
UTC+5
UTC+6
UTC+7
UTC+8
UTC+9
UTC+10
UTC+11
UTC+12
UTC+13
UTC+14